

This is Google's cache of <http://opensoundcontrol.org/topic/42>. It is a snapshot of the page as it appeared on 24 Jan 2016 23:56:33 GMT.

The [current page](#) could have changed in the meantime. [Learn more](#)

[Full version](#) [Text-only version](#) [View source](#)

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac) and use the find bar.

[Introduction to OSC](#)[OSC 1.0 Specification](#)[OSC 1.1 Specification](#)[Application Areas](#)[Developer Resources](#)[Forums](#)[Implementations](#)[Publications](#)**opensoundcontrol.org**[Home](#)

UDP Spoofing and DOS attacks

November 15, 2005 - 3:04pm — Adrian Freed

Most internet uses of OSC use UDP which is trivial to spoof. This and the potential expense of a message constructed with a broad matching pattern mean that OSC applications are susceptible to Denial of Service attacks.

The easiest workaround is to use TCP which is harder to spoof. Can anyone help develop the appropriate magic to make the UDP implementation more robust?

[Login to post comments](#)[Transport](#)

Challenge-response is not necessarily stateful.

August 8, 2006 - 10:19am — aws

TCP alone is not a workaround because it does not ensure endpoint identification, nor is it tamper-proof. The easiest and most secure way to ensure security for an OSC connection is via a secure tunnel, e.g. VPN, SSH tunneling etc.

If one wanted to expose a UDP-OSC port to the open internet *without* using a secure tunnel there would be several considerations:

1. Some form of authentication for client and server identification which is reasonably safe over an insecure transport
2. Some form of identification to prevent spoofed replies (e.g. ala the DNS ID method or challenge-response)
3. Rate-limiting to prevent overload of the server and/or network
4. Limited number of failed authentication attempts before a client is blocked

Assuming that we are not doing our online banking over OSC, challenge-response authentication is a good way to go, and its reasonably secure. It relies on a shared-secret (i.e. password) already known on both ends. Additionally it is not actually a stateful protocol! Typically the challenge is a randomly generated string produced by the server, then returned by the client along with a crypted signature of the challenge+password. However without any loss in security the string can be generated by the client, making it a stateless protocol. A couple of caveats; 1) the client must be trusted to have a good random number generator and 2) the cryptographic hash algorithm should be strong enough to keep the secret unguessable even under repeated exposure (MD5 probably does not qualify).

Example; force all packets to be bundles, with the first encapsulated message for authentication.

User login

Username: *

Password: *

☐ Remember me

[Request new password](#)

Navigation

[Recent posts](#)

New Publications

[Improving the Efficiency of Open Sound Control with Compressed Address Strings](#)

[Best Practices for Open Sound Control](#)

[Minuit : Propositions for a query system over OSC](#)

[micro-OSC: The Open Sound Control Reference](#)

[Implementation for Embedded Devices](#)

[c](#)

Who's online

There are currently *0 users* and *11 guests* online.

```
[ #bundle timestamp ( auth-message, message, message, ... ) ]
```

The auth message will be something like this:

```
[ /crauth challenge-string response-string ]
```

The challenge-string is randomly generated on every packet, and response-string = hash(timestamp + other-messages-content + challenge-string + shared-secret) (where hash = SHA1, MD5 etc). Including the contents of the other messages in the hash prevents tampering of the the contents.

This goes both ways, i.e. the server authenticates itself to the client as well using the same scheme.

For applications where there is no shared secret but you want to prevent spoofed replies, one could use the unique ID method that is used in DNS (this is described well in many articles on DNS spoofing and its prevention).

[Login](#) to post comments

c

Copyright The Center For New Music and Audio Technology (CNMAT), UC Berkeley
Copyright for comments and postings are the property of the respective author.

[Main Page RSS Feed](#) | [All Forum Comments RSS Feed](#)